Remarks

Applicant respectfully requests reconsideration of this application as amended. Claims 1, 3, 5-6, 9, 11, 13, and 17-22 have been amended. Claims 2, 4, 8, and 10 have been cancelled. No claims have been added. Therefore, claims 1, 3, 5-7, 9, and 11-22 are presented for examination.

35 U.S.C. §112 Rejection

Claims 3, 9, and 17 stand rejected under 35 U.S.C. §112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention. More specifically, the Office Action states that "'CLI' in claims 3, 9, and 17 is used by the claim to mean 'a programming language', while the accepted meaning is 'an operating system.'" (Office Action at pg. 1, point 4.) The term "CLI" has been removed from claims 3, 9, and 17. Therefore, applicant respectfully requests the 35 U.S.C. §112 rejection be withdrawn.

35 U.S.C. §101 Rejection

Claims 1-6 and 18-22 stand rejected under 35 U.S.C. §101 because the claimed invention is directed to non-statutory subject matter. First, the Office Action states that the language of claim 1 raises a question as to whether the claim is directed merely to an abstract idea. (Id. at pg. 3, point 7.) Claims 1-6 have been amended to recite a "computer-implemented method". Applicant submits that claims 1-6 are directed to statutory subject matter. Therefore, applicant respectfully requests the withdrawal of the 35 U.S.C. §101 rejection against claims 1-6.

Second, the Office Action states that in claim 18, the claimed "instruction set" is not statutory for at least the reason that it is not tangibly embodied in a manner as to be executable. Claims 18-22 have been amended to recite a "computer system-executable instruction set." Applicant submits that claims 18-22 are directed to statutory subject matter. Therefore, applicant respectfully requests the withdrawal of the 35 U.S.C. §101 rejection against claims 18-22.

35 U.S.C. §102(b) Rejection

Claims 1-22 stand rejected under 35 U.S.C. §102(b) as being anticipated by Hudson et al. ("Cycles to Recycle: Garbage Collection on the IA-64"). Applicant submits that the present claims are patentable over Hudson.

Hudson is a symposium paper describing Intel's 64-bit instruction set architecture's features relating to support of garbage collection (GC). The paper describes how one may exploit features of the IA-64. It provides object scanning techniques, offering techniques for object allocation and the Java "jsr problem". It also discusses ordering memory accesses and how the IA-64 achieves publication safety efficiently. (Hudson at Abstract.)

Claim 1, as amended, recites:

> A computer-implemented method comprising:
> monitoring thread switches in a multiple-threaded application through use of a single thread switch flag;
> executing a non-blocking thread synchronization sequence; and
> interrupting the non-blocking thread synchronization sequence upon the occurrence of a thread switch; and
> repeating the non-blocking thread synchronization sequence, wherein the non-blocking thread synchronization sequence is idempotent in order to abandon the non-blocking thread synchronization sequence in mid-sequence without consequences.

First, applicant submits that Hudson is improperly applied as a prior art reference under 35 U.S.C. §102(b). Referring to Exhibit A included with this response, the Hudson reference was published as part of The 2000 International Symposium on Memory Management. Exhibit A illustrates that this Symposium and corresponding publication of the Hudson reference occurred on October 16, 2000. (See pgs. 1 & 4.) The filing date of the present application occurred on October 12, 2001. Therefore, the Hudson reference occurred less than one year prior to the filing date of the present application and is improperly applied under 35 U.S.C. §102(b). As such, applicant respectfully requests the modification of the application of 35 U.S.C. §102 rejection to the present application.

Second, applicant submits that Hudson does not disclose or suggest monitoring thread switches in a multiple-threaded application through use of a single thread switch flag, as recited by claim 1. Hudson requires the use of a ***pair of predicate registers*** to indicate that a task switch has occurred. (Hudson at §4.2.2.) This is not the same as utilizing a single thread switch flag to monitor thread switches in a multi-threaded application, as recited in claim 1.

Third, applicant submits that Hudson does not disclose or suggest repeating the non-blocking thread synchronization sequence, wherein the non-blocking thread synchronization sequence is idempotent in order to abandon the non-blocking thread synchronization sequence in mid-sequence without consequences, as recited by claim 1. As an initial matter, applicant can find no disclosure or suggestion in Hudson of a non-blocking thread synchronization sequence. Figure 14 of Hudson shows a common case of a synchronization implementation strategy known as thin locks. Yet, as indicated in Hudson, this strategy utilizes locks to protect resources. (Id. at §4.8.) This is not the same as a ***non-blocking*** thread synchronization sequence, as recited by claim 1. Furthermore, applicant can find no disclosure or suggestion in Hudson of the non-blocking thread synchronization sequence being idempotent in order to abandon the non-blocking thread synchronization sequence in mid-sequence without consequence.

Therefore, for the reasons discussed above, claim 1 is patentable over Hudson. Claims 3 and 5-6 depend from claim 1 and include additional limitations. As a result, claims 3 and 5-6 are also patentable over Hudson.

Independent claims 7, 13, and 18 also recite, in part, monitoring thread switches in a multiple-threaded application through use of a single thread switch flag and repeating the non-blocking thread synchronization sequence, wherein the non-blocking thread synchronization sequence is idempotent in order to abandon the non-blocking thread synchronization sequence in mid-sequence without consequences. As discussed above Hudson does not disclose or suggest such a feature. Therefore, claims 7, 13, and 18, as well as their respective independent claims, are patentable over Hudson for the reasons discussed above with respect to claim 1.

Applicant respectfully submits that the rejections have been overcome and that the claims are in condition for allowance. Accordingly, applicant respectfully requests the rejections be withdrawn and the claims be allowed.

The Examiner is requested to call the undersigned at (303) 740-1980 if there remains any issue with allowance of the case.

Applicant respectfully petitions for an extension of time to respond to the outstanding Office Action pursuant to 37 C.F.R. § 1.136(a) should one be necessary. Please charge our Deposit Account No. 02-2666 to cover the necessary fee under 37 C.F.R. § 1.17(a) for such an extension.

Please charge any shortage to our Deposit Account No. 02-2666.

Respectfully submitted,

BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN LLP

Date: March 22, 2006

Ashley Ott

Ashley R. Ott
Reg. No. 55,515

12400 Wilshire Boulevard
7th Floor
Los Angeles, California 90025-1026
(303) 740-1980

Application No.09/977,509

Exhibit A

The 2000 International Symposium on Memory Management

Regal Minneapolis Hotel
Minneapolis, Minnesota, USA
15-16 October, 2000

Sponsored by
ACM SIGPLAN
Co-located with
OOPSLA 2000

Registration
Programme
Committee
Links

Call for Participation and Advance Programme

The International Symposium on Memory Management is a forum for research in memory management, in all its diversity. Areas of interest include but are not limited to: garbage collection, dynamic storage allocation, storage management implementation techniques and their interactions with languages and operating systems, and empirical studies of programs' memory allocation and referencing behavior.

ISMM 2000 continues the tradition of the successful conference series established with the International Workshops on Memory Management held in 1992 (St. Malo, France) and 1995 (Kinross, Scotland), and the inaugural ISMM in 1998 (Vancouver, Canada). Proceedings of the IWMM conferences are available from Springer-Verlag (Lecture Notes in Computer Science no. 637 and no. 986), while the ISMM'98 proceedings were published by the ACM. The 2000 Symposium Proceedings are also published by the ACM.

Registration

Please register online via https://www.cmsusa.com/ismm/. Registration includes lunches, entry to the OOPSLA tutorial/workshop reception and the conference dinner on Sunday, and entry to the OOPSLA welcome reception on Monday. Information on accomodation arrangements, and the conference location at the Regal Minneapolis

Hotel, is available via the OOPSLA web-site.

Programme

	Sunday October 15th		**Monday October 16th**
8:15-9:00	Coffee & Registration*	8:15-9:00	Coffee
9:00-9:15	Welcome & Introduction	9:00-10:20	Session III: *HardwareSupport*
9:15-10:20	Invited Speaker: Jon L. White, *CommerceOne, Inc.*		
10:20-10:50	Break	10:20-10:50	Break
10:50-12:30	Session I: *Accuracy & Locality*	10:50-12:30	Session IV: *Profiling & Object Lifetimes*
12:30-14:00	Lunch	12:30-14:00	Lunch
14:00-15:00	Proposal & Discussion: *Trace Formats*	14:00-15:00	Informal Presentations
15:00-15:30	Break	15:00-15:30	Break
15:30-17:00	Session II: *Implementation*	15:30-17:00	Session V: *Concurrent & Distributed*
17:30-18:30	OOPSLA tutorial/workshop reception	17:30-19:30	OOPSLA welcome reception
18:30-20:30	Conference dinner		

*Registration will also be available from 7am to 5pm on Sunday, and 7am to 2pm on Monday.

Invited Speaker: *Jon L. White,* CommerceOne, Inc

Sunday 15th October, 9:15-10:20

Conquering the Hurdles, Challenging the Horizons A Historical Perspective on Memory Management

We will travel through a retrospective overview of some of the great "hurdles" in the area of Memory Management, from the discovery of "Garbage Collection", through the realization of the potential harmfulness of GC, to the expanded horizons which are driven by paradigm shifts in application programs. In the process, we will review some of the very early approaches to memory management to appreciate just how far we have come, and just how much the great leaps of this science have been driven by the appearance of cheaper and faster hardware. We will also re-visit some "war stories" of the past where a paradigm shift was forced because of unforseen consequences in what was otherwise perceived to be a "Very Good Idea(tm)". Examples will include the speaker's own pioneering experiences in shifting from pure "collection" techniques to BIBOP storage strategies, compile-time change-of-representation for a one to two orders of magnitude speedup in numerical programs written in Lisp, to the discovery of the "Pig-in-a-Python" syndrome. Speculative, overview-like conjectures will be made about the current horizons of the field, especially with respect to large databases and the Java programming language.

A Brief Biography of Jon L. White

JonL received a B.S. degree in Mathematics from Carnegie-Mellon University, and an M.A. in Applied Mathematics from Harvard University. He subsequently joined Marvin Minsky's team at the M.I.T Artifical Intelligence Laboratory, where he began the development of high-quality Lisp systems and compilers. The PDP10 MacLisp efforts subsequently supported not only the A.I. lab, but the MACSYMA Symbolic Algegra group, and ultimately many other university and commercial research labs. JonL also spent one calendar year away from MIT at IBM's Watson Research Center in Yorktown Heights, NY, contributing to the development of Lisp/370.

At the birth of the commercial A.I. revolution, JonL joined a group at Xerox's Palo Alto Research Center tasked with product development of Interlisp on the special-purpose, microded Xerox D-series machines. A few years later, in January 1985, he joined the "startup" company Lucid, Inc. whose goal was to make high-quality Common Lisp systems available on all the prevalent "stock" hardware machines. As the commercial A.I. market - and consequently the commercial Common Lisp market - went into decline in the early 1990's, JonL joined the Lisp group at Harlequin. In all of these commercial Lisp developments, JonL contributed substantially not only to the compiler development, but also to both "GC" and higher-level memory management.

After a brief 8 months at NASA/Ames Research Center working with the Automated Software Engineering group there, JonL joined yet another startup - a spin-off from Stanford University - which was quickly bought out by his present employer CommerceOne. He is currently working on a product called iMerge, written in Lisp and a result of many years of A.I. research at Stanford which is directed at the database needs of today's Business-to-Business commercial environment.

Session I: *Accuracy & Locality*

Sunday 15th October, 10:50-12:30
Chair: Greg Morrisett
10:50-11:20

On the Type Accuracy of Garbage Collection, Martin Hirzel and Amer Diwan

11:20-11:40

On the Effectiveness of GC in Java, Ran Shaham, Elliot Kolodner and Mooly Sagiv

11:40-12:00

Thread-Specific Heaps for Multi-Threaded Programs, Bjarne Steensgaard

12:00-12:30

A Region-Based Memory Manager for Prolog, Henning Makholm

Proposal & Discussion: Trace Formats

Sunday 15th October, 14:00-15:00

Designing a Trace Format for Heap Allocation Events, Trishul Chilimbi, Richard Jones and Benjamin Zorn

Session II: *Implementation*

Sunday 15th October, 15:30-17:00

15:30-16:00

Compact Garbage Collection Tables, David Tarditi

16:00-16:20

Reducing Garbage Collector Cache Misses, Hans-J. Boehm

16:20-16:40

Memory Allocation with Lazy Fits, Yoo C. Chung and Soo-Mook Moon

16:40-17:00

Conservative Garbage Collection for General Memory Allocators, Gustavo Rodriguez-Rivera, Mike Spertus and Charles Fiterman

Session III: *Hardware Support*

Monday 16th October, 09:00-10:20

09:00-09:30

Concurrent Garbage Collection Using Hardware-Assisted Profiling, Timoth H. Heil and James E. Smith

09:30-09:50

Concurrent Garbage Collection Using Program Slices on Multithreaded Processors, Manoj Plakal and Charles N. Fischer

09:50-10:20

Cycles to Recycle: Garbage Collection on the IA-64, Richard L. Hudson, J. Eliot B. Moss, Sreenivas Subramoney and Weldon Washburn

Session IV: *Profiling & Object Lifetimes*

Monday 16th October, 10:50-12:30

Chair: Trishul Chilimbi

10:50-11:20

The Case for Profile-Directed Selection of Garbage Collectors, Robert Fitzgerald and David Tarditi

11:20-11:40

Efficient Object Sampling via Weak References, Ole Agesen and Alex Garthwaite

11:40-12:10

Dynamic Adaptive Pre-Tenuring, Timothy L. Harris

12:10-12:30

On Models for Object Lifetimes, Darko Stefanovic, Kathryn S. McKinley and J. Eliot B. Moss

Session V: *Concurrent & Distributed*

Monday 16th October, 15:30-17:00

15:30-16:00

A Generational Mostly-Concurrent Garbage Collector, Tony Printezis and David Detlefs

16:00-16:30

Implementing an On-the-Fly Garbage Collector for Java, Tamar Domani, Elliot K. Kolodner, Ethan Lewis and Eliot E. Salant

16:30-17:00

Diffusion Tree Restructuring for Indirect Reference Counting, Peter Dickman

Conference Committee

General Chair

Craig Chambers
Department of Computer Science and Engineering
University of Washington
Box 352350
Seattle, WA 98195-2350 USA
+1 (206) 685-2094; fax: +1 (206) 543-2969
chambers@cs.washington.edu

Programme Chair

Tony Hosking
Department of Computer Sciences
Purdue University
1398 Computer Science Building
West Lafayette, IN 47907-1398 USA
+1 (765) 494-6001; fax: +1 (765) 494-0739
hosking@cs.purdue.edu

Steering Committee

Hans Boehm
Hewlett-Packard

Richard Jones
University of Kent at Canterbury, UK

J. Eliot B. Moss
University of Massachusetts at Amherst, USA

Simon Peyton Jones
Microsoft Research, UK

Ben Zorn
Microsoft Research, USA

Programme Committee

Trishul Chilimbi
Microsoft Research, USA

Bart Demoen
Katholieke Universiteit Leuven, Belgium

David Detlefs
Sun Microsystems Laboratories, USA

Eric Jul
University of Copenhagen, Denmark

Greg Morrisett
Cornell University, USA

Scott Nettles
University of Texas at Austin, USA

Application No. 09/977,509

Marc Shapiro
Microsoft Research, UK

Related Web Pages

Previous ISMM and IWMM proceedings

- ISMM'98
- IWMM'95
- IWMM'92

Other resources

- Richard Jones's GC page (contains extensive GC bibliography and links to related resources)
- Wilson et al.'s OOPS Group Home Page (contains extensive surveys on garbage collection and memory allocators)
- David Chase et al.'s GC FAQ
- Harlequin's Memory Management Reference page
- Collected papers from the (unrefereed) OOPSLA workshops on garbage collection and memory management, from 1990, 1991, 1993 and 1997.

Maintained by Richard Jones